Listas Duplamente Encadeadas

Allan Robson

DCA

Disciplina: Estruturas de Dados

07/03/2017

Listas Encadeadas

Listas Encadeadas

Listas

- Uma estrutura do tipo **Lista** é uma sequência de elementos do mesmo tipo.

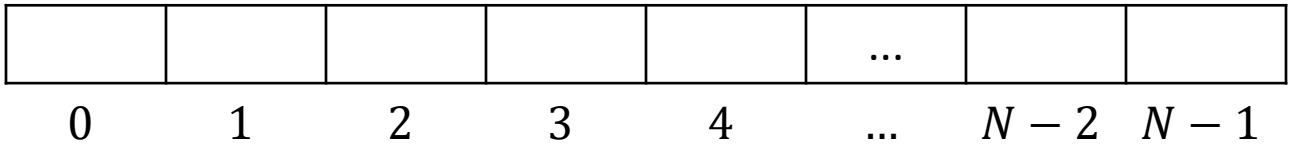
- Seus elementos possuem estrutura interna abstraída, ou seja, sua complexidade é arbitrária e não afeta o seu funcionamento.

33	23	16	15	43	...

LISTAS ENCADEADAS

Listas

- Implementação:
  - Sequencial (estática);

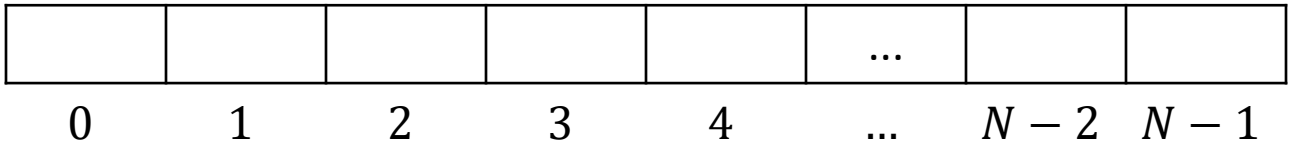


  - Encadeada (dinâmica)

LISTAS ENCADEADAS

Alocação Sequencial X Alocação Encadeada

- Alocação Sequencial (estática):
  - Faz uso de *arrays* para armazenar os dados;
  - Possui **tamanho fixo** (alocação sequencial);
  - Dados ocupam **posições contíguas na memória**;
  - Depois que a quantidade máxima de elementos é alcançada, só é permitido adicionar um novo elemento caso algum elemento seja removido;



LISTAS ENCADEADAS

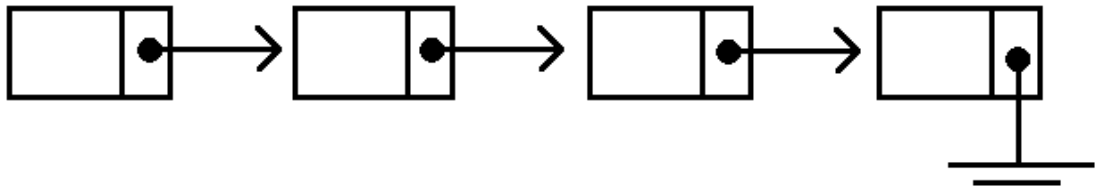
Alocação Sequencial X Alocação Encadeada

- Alocação Encadeada (dinâmica):
  - NÃO possui tamanho fixo (alocação dinâmica);
  - Os nós de uma lista encontram-se aleatoriamente disposto na memória e são **interligados por ponteiros**, que indicam a posição do próximo elemento.

LISTAS ENCADEADAS

Listas Simplesmente Encadeadas

- É composta por **nós que apontam para o próximo elemento** da lista, o último elemento apontará para nulo.
- Para compor uma lista encadeada, basta guardar seu primeiro elemento.
- Para acessar um elemento, é preciso percorrer todos seus antecessores na Lista.

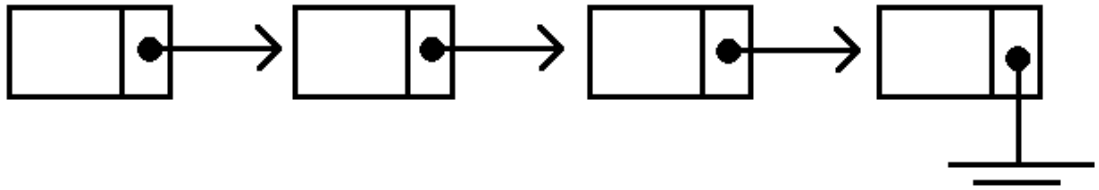


Listas Encadeadas

Listas Simplesmente Encadeadas

- Vantagens:
  - Melhor utilização dos recursos de memória;
  - Não precisa movimentar os elementos nas operações de inserção e remoção;

- Desvantagem:
  - Acesso indireto aos elementos;
  - Necessidade de percorrer a lista para acessar um elemento;

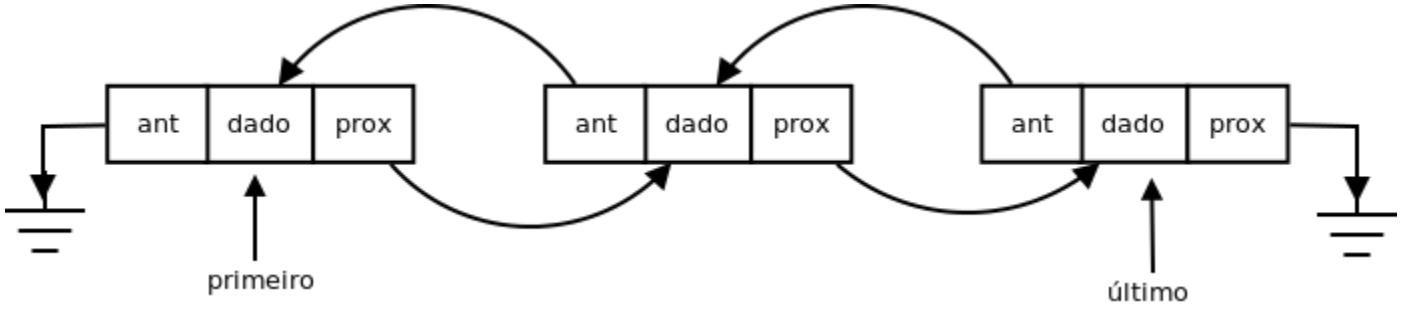


Listas Duplamente Encadeadas

Listas Duplamente Encadeadas

Definição

- Cada elemento da lista aponta para seu **sucessor** e seu **antecessor** na lista.



Listas Duplamente Encadeadas

Definição

- Quando vou utilizar uma Lista Duplamente Encadeada?
  - **Necessidade de acessar informação de um elemento antecessor.**

- Aplicações:
  - Funcionalidade de desfazer (ctrl+z) dos programas;
  - O cache do browser que permite voltar para páginas visitadas anteriormente;
  - Player de músicas: avançar e voltar músicas.

Listas Duplamente Encadeadas

Operações

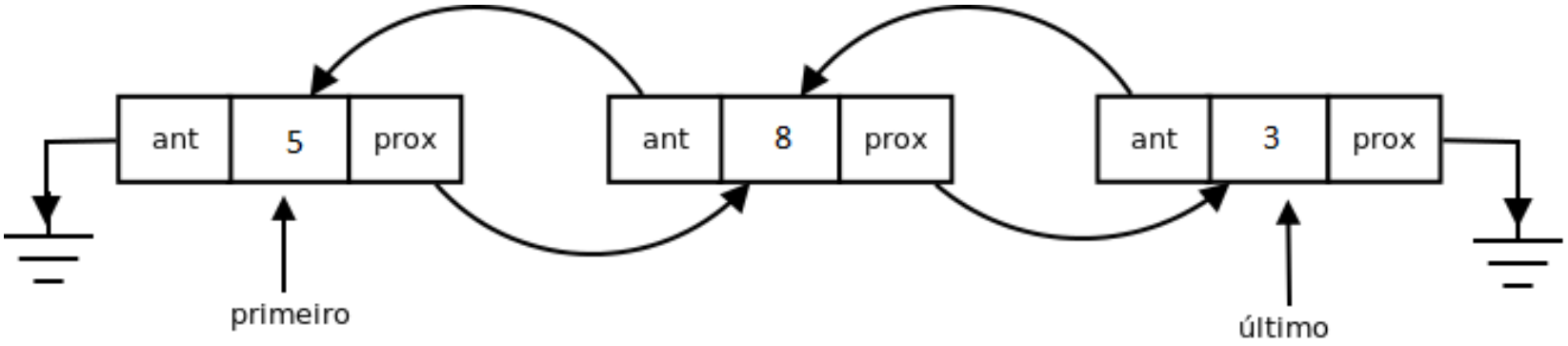
- As principais operações utilizadas quando trabalhamos com Listas são:
  - Busca
  - Inserção
  - Remoção

Listas Duplamente Encadeadas

Operações: Busca

- Para buscar um determinado elemento em uma lista encadeada é necessário realizar uma **Busca Linear**;

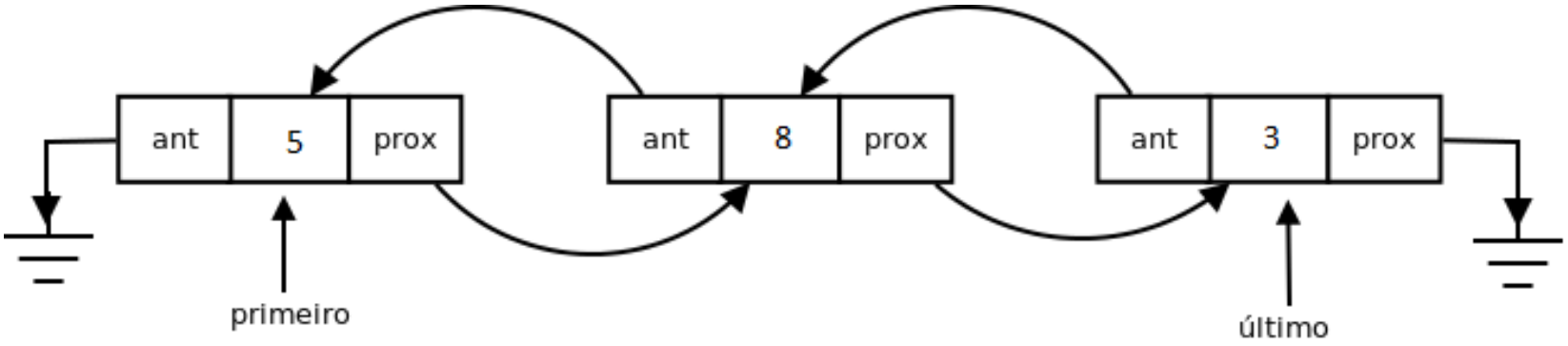
- A busca é iniciada a partir do primeiro elemento da lista;



Listas Duplamente Encadeadas

Operações: Busca

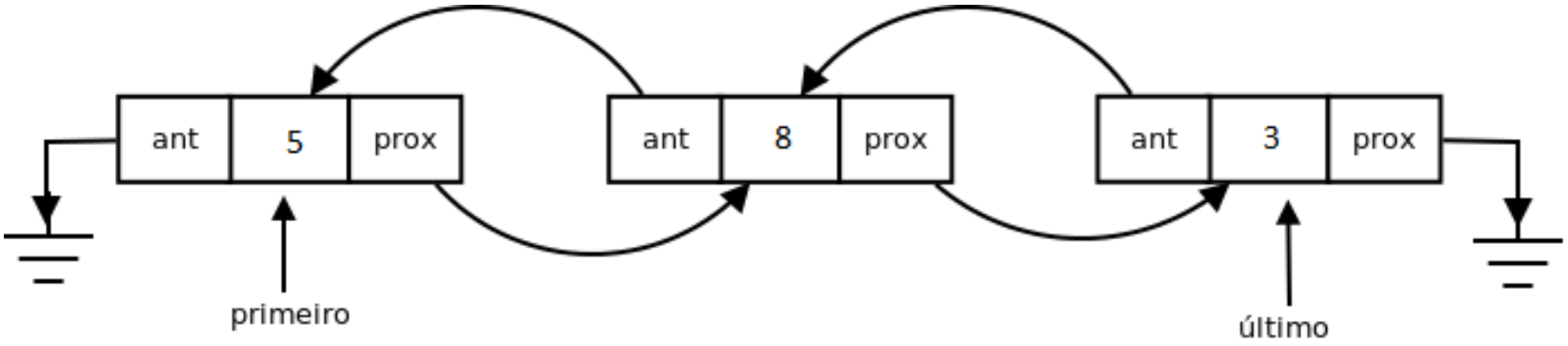
- Exemplo: Desejamos buscar o elemento ‘3’ na lista.



Listas Duplamente Encadeadas

Operações: Busca

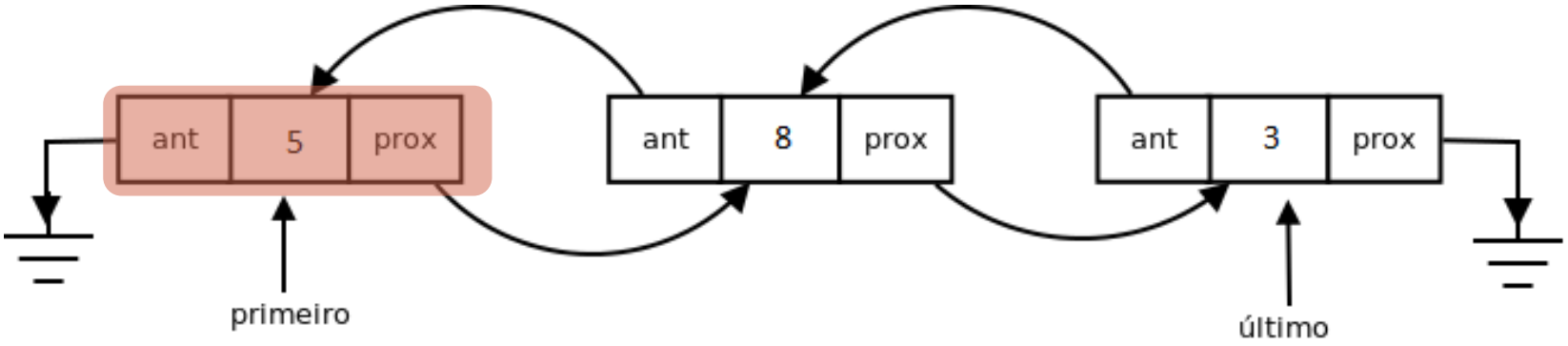
- Exemplo: Desejamos buscar o elemento ‘3’ na lista.
  - Primeiro verificamos se a lista está vazia ou não;



Listas Duplamente Encadeadas

Operações: Busca

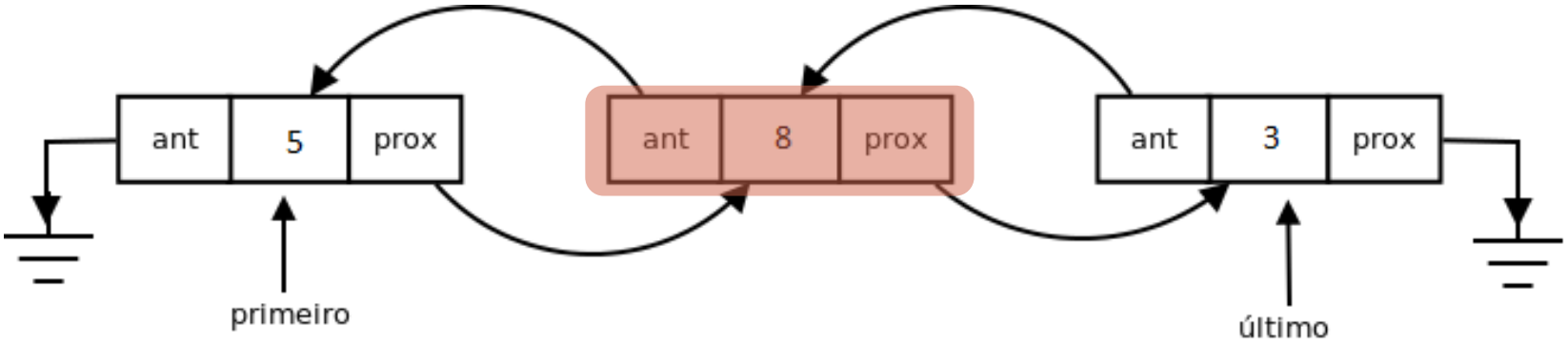
- Exemplo: Desejamos buscar o elemento ‘3’ na lista.
  - Como a lista não está vazia, verificamos se o primeiro elemento da lista é o elemento procurado.



Listas Duplamente Encadeadas

Operações: Busca

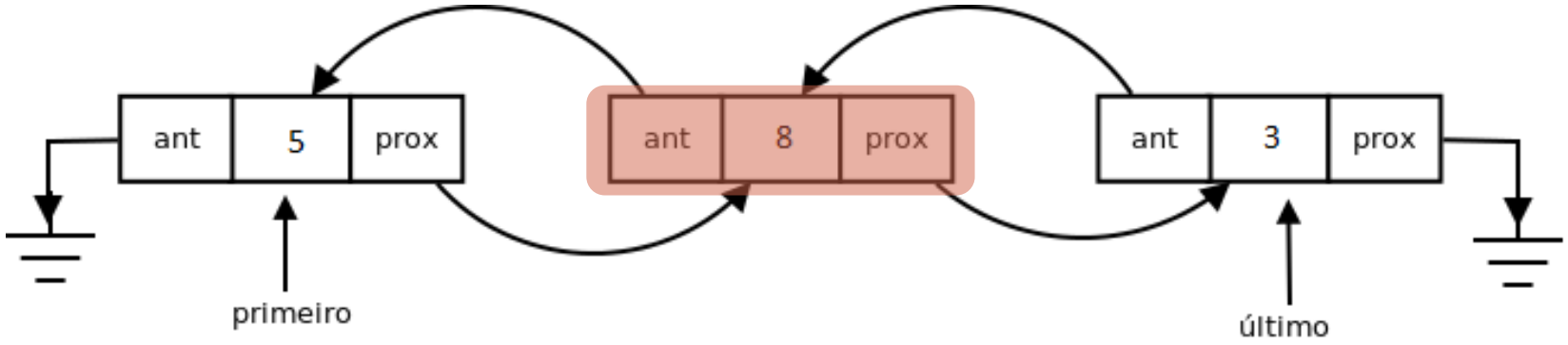
- Exemplo: Desejamos buscar o elemento ‘3’ na lista.
  - O elemento procurado NÃO é o primeiro, então a busca prossegue para o segundo elemento.



Listas Duplamente Encadeadas

Operações: Busca

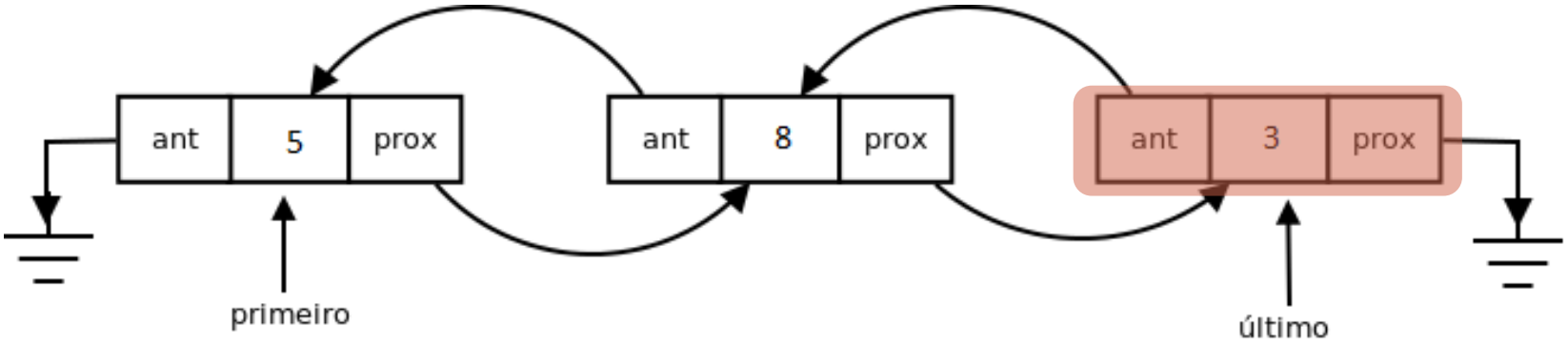
- Exemplo: Desejamos buscar o elemento ‘3’ na lista.
  - O elemento procurado NÃO é o primeiro, então a busca prossegue para o segundo elemento.
  - Novamente o elemento procurado não foi encontrado. Então a busca prossegue para o elemento seguinte.



Listas Duplamente Encadeadas

Operações: Busca

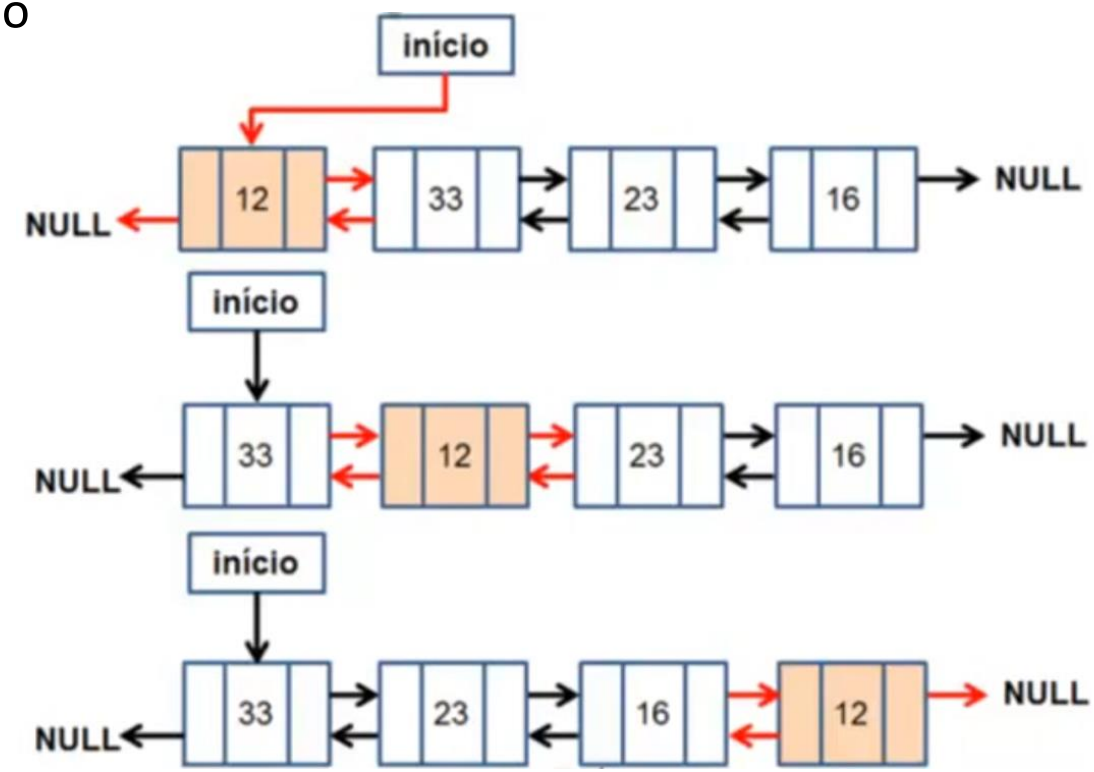
- Exemplo: Desejamos buscar o elemento ‘3’ na lista.
  - O elemento procurado foi encontrado.



Listas Duplamente Encadeadas

Operações: Inserção

- Existem três tipos de **inserção**:
  - Inicio
  - Meio
  - Fim

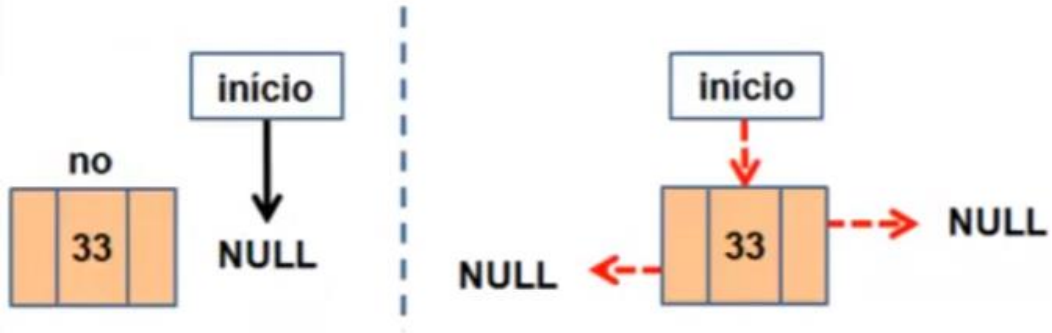


Listas Duplamente Encadeadas

Operações: Inserção

- Caso particular: **Lista Vazia**
  - Antes de inserir um elemento é necessário verificar se a lista está vazia ou não.

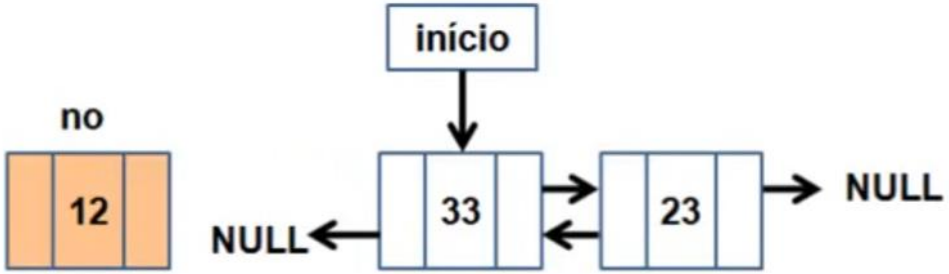
```
if (inicio == NULL) {
    inicio = no;
    inicio.ant = NULL;
    inicio.prox = NULL;
}
```



Listas Duplamente Encadeadas

Operações: Inserção no Início

- **Inserção no Início:**
  - Deseja-se inserir o elemento 12 na lista abaixo:

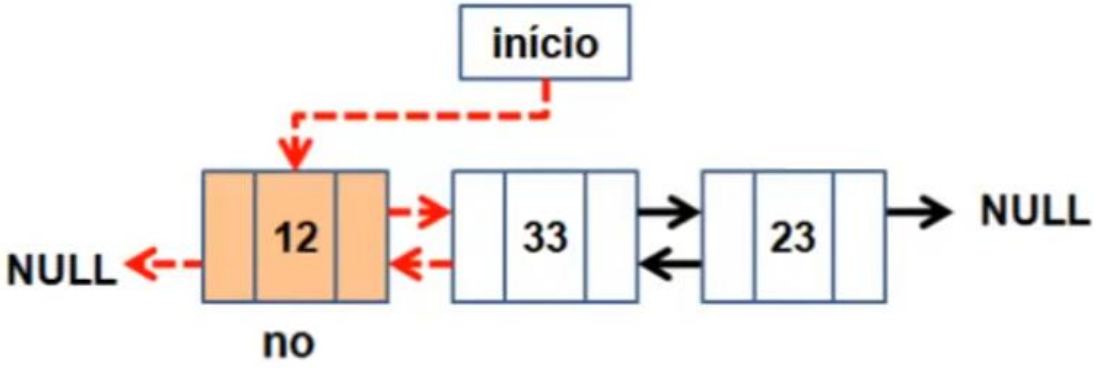


Listas Duplamente Encadeadas

Operações: Inserção no Início

- **Inserção no Início:**
  - Deseja-se inserir o elemento 12 na lista abaixo:
  - Sequência se passos:

```
no.prox = inicio;
no.ant = NULL;
inicio.ant = no;  (Lista não vazia)
inicio = no
```

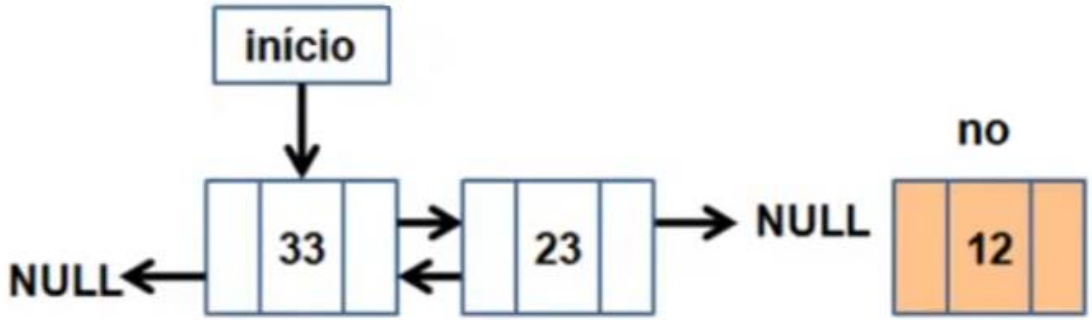


Listas Duplamente Encadeadas

Operações: Inserção no Final

- **Inserção no Final:**
  - É necessário **percorrer a lista** antes de inserir o novo elemento;

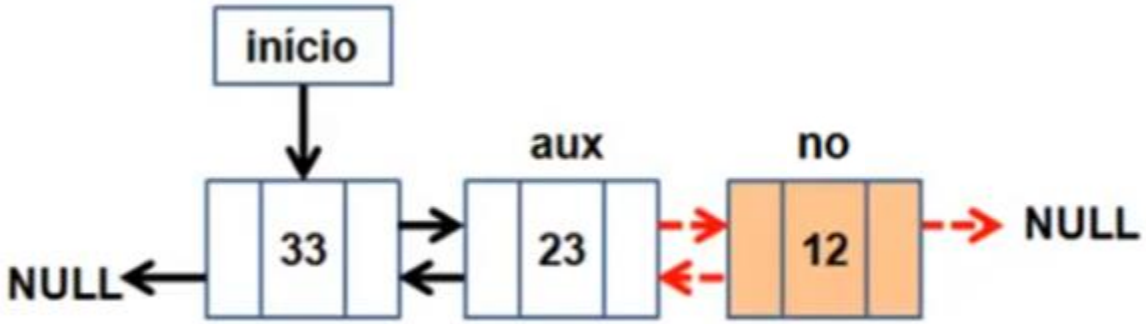
```
aux = inicio;
while (aux.prox != NULL) {
     aux = aux.prox;
}
```



Listas Duplamente Encadeadas

Operações: Inserção no Final

- **Inserção no Final:**
  - Após percorrer toda a lista, inserimos o novo elemento após o último elemento (`aux`);
    - **`no.prox = NULL;`**
    - **`no.ant = aux;`**
    - **`aux.prox = no;`**



Listas Duplamente Encadeadas

Operações: Inserção no Meio

- **Inserção no Meio:**
  - Normalmente utilizado em inserções de forma **ordenada**;
  - Novamente é necessário **percorrer a lista** e procurar onde o novo elemento será inserido (entre os ponteiros '*ante*' e '*atual*');

```
ante = NULL; atual = inicio;
while (atual != NULL &&
            atual.valor<novoValor) {
    ante = atual;
    atual = atual.prox;
}
```

- **Inserção no Meio:**
  - Após encontrar a posição onde o elemento será inserido, algumas situações podem acontecer:
    - **Inserir no início**:

```
if (atual == inicio) {
        no.ant = NULL;
        inicio.ant = no;
        no.prox = inicio;
        inicio = no;
}
```

Listas Duplamente Encadeadas

Operações: Inserção no Meio

- **Inserção no Meio:**
  - Após encontrar a posição onde o elemento será inserido, algumas situações podem acontecer:
    - **Inserir no meio ou final**:

```
else {
        no.prox = atual;
        no.ant = ante;
        ante.prox = no;
        if (atual != NULL) {
            //Não é final da lista
            atual.ant = no;
        }
}
```

Listas Duplamente Encadeadas

Operações: Remoção

- Existem três tipos de **remoção**:
  - Início
  - Meio
  - Fim

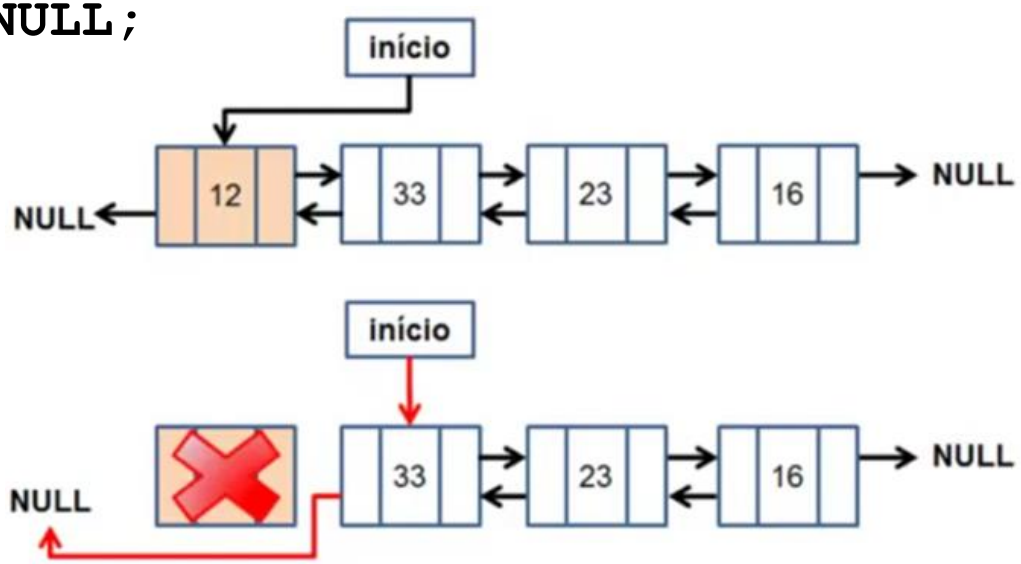
- Cuidados:
  - Não é possível remover elemento de uma **lista vazia;**
  - Ao remover o **último elemento** de uma lista, a mesma ficará vazia.

Listas Duplamente Encadeadas

Operações: Remoção

- **Remoção no Início:**

```
no = inicio;
inicio = inicio.prox;
if (inicio != NULL) {
   //Lista com mais de um elemento
   inicio.ant = NULL;
}
free(no)
```

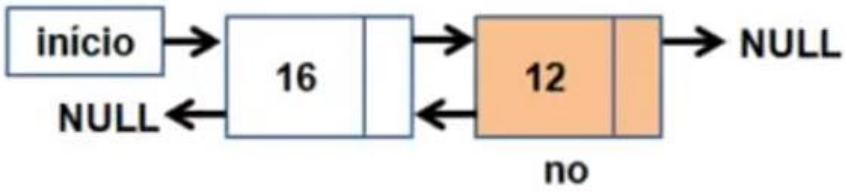


Listas Duplamente Encadeadas

Operações: Remoção

- **Remoção no Final:**
  - Antes de remover, procura-se o último elemento da lista.

```
no = inicio;
while(no.prox != NULL) {
    no = no.prox;
}
```

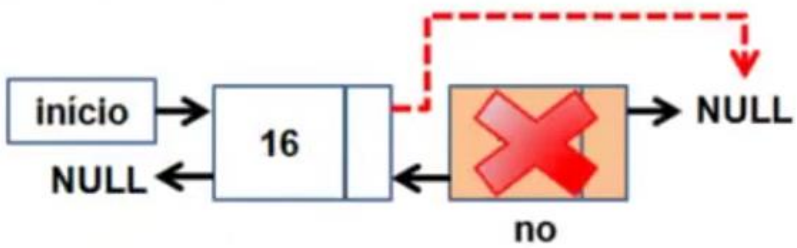


Listas Duplamente Encadeadas

Operações: Remoção

- **Remoção no Final:**
  - Após percorrer a lista, remove-se o último elemento.

```
if (no.ant == NULL) {
    //Lista com apenas 1 elemento
    inicio = NULL;
} else {
    no.ant.prox = NULL;
}
free(no)
```

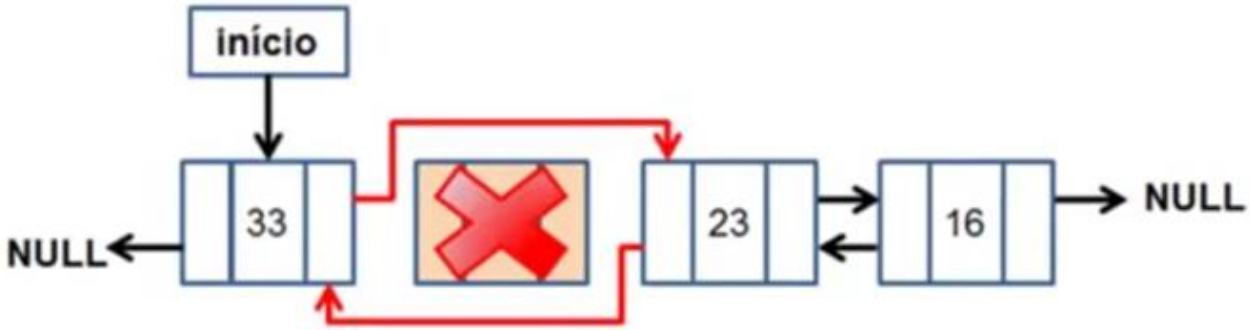


Listas Duplamente Encadeadas

Operações: Remoção

- **Remoção no Meio:**
  - Primeiro é feito uma busca pelo elemento que deseja-se remover.

```
no = inicio;
while(no != NULL && no.valor!=elemRemover) {
    no = no.prox;
}
```



Listas Duplamente Encadeadas

Operações: Remoção

- **Remoção no Meio:**
  - Em seguida remove-se o elemento, caso tenha sido encontrado.

```
if (no == NULL) {
    return erro; //Elemento não encontrado
}
if (no->ant == NULL) {
    //Remover o primeiro elemento
    inicio = no.prox;
} else {
    no.ant.prox = no.prox;
}
if (no.prox != NULL) {
    //Não é o último elemento
    no.prox.ant = no.ant;
}
free(no)
```

Listas Duplamente Encadeadas

Operações: Remoção

- **Remoção no Meio:**
  - Em seguida remove-se o elemento, caso tenha sido encontrado.